CME 305: Discrete Mathematics and Algorithms

TA: Kun Yang (kunyang@stanford.edu)

1. Consider an optimization version of the Hitting Set Problem defined as follows. We are given a set $A = \{a_1, ..., a_n\}$ and a collection $B_1, ..., B_m$ of subsets of A. Also, each element $a_i \in A$ has a weight $w_i \geq 0$. The problem is to find a hitting set $H \subseteq A$ such that the total weight of the elements in H, that is, $\sum_{a_i \in H} w_i$, is as small as possible (H is hitting set if $H \cap B_i$ is not empty for each i). Let $b = \max_i |B_i|$ denote the maximum size of any of the sets $B_1, ..., B_m$. Give a polynomial-time approximation algorithm for this problem that finds a hitting set whose total weight is at most b times the minimum possible.

   Solutions. Consider the following LP (the total number of elements is n),

$$\min \sum_{i=1}^{n} w_i x_i$$

$$s.t. \quad 0 \leq x_i \leq 1, i = 1, ..., n; \sum_{i:a_i \in B_j} x_i \geq 1, j = 1, ..., m$$

   Let x be the solution of this problem, and w_{LP} is the optimal value.

   Now define the set S to be all those elements where $x_i \geq 1/b$, i.e., $S = \{a_i | x_i \geq 1/b\}$

   (1) S is a hitting set. In fact, we know that the sum of all x_i where $a_i \in B_j$ is at least 1. The set B_j contains at most b elements. Therefore some $x_i \geq 1/b$, for some $a_i \in B_j$. By definition of S, the element $a_i \in B$. So B_j intersects with S by a_i.

   (2) The total weight of all elements in S is at most bw_{LP}: for each $a_i \in S$ we know that $x_i \geq 1/b$, i.e., $1 \leq bx_i$, therefore

$$w(S) = \sum_{a_i \in S} w_i \leq \sum_{a_i \in S} w_i b x_i \leq b \sum_{i=1}^{n} w_i x_i = b w_{LP}$$

   (3) S^* is the optimal set and let $x_i^* = 1$ if $a_i \in S^*$ and $x_i^* = 0$ otherwise. Then the vector x^* satisfies the LP above. Thus,

$$w_{LP} \leq \sum_{i=1}^{n} w_i x_i^* = \sum_{a_i \in S} w_i = w(S^*)$$

   Therefore we have a hitting set S, such that $w(S) \leq b w(S^*)$

2. DNA Fragment Assembly: assemble individual short fragments (reads) into a single genomic sequence ("superstring").

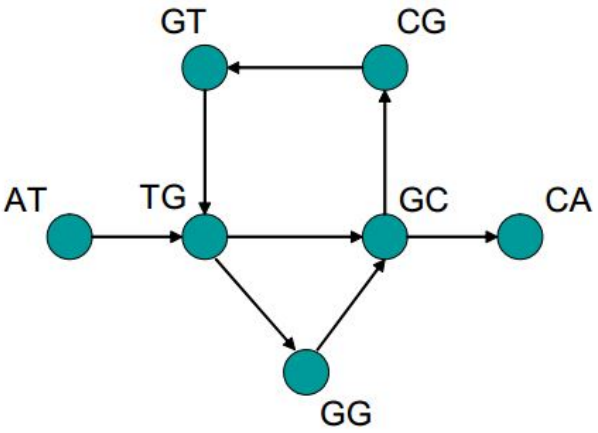
   Problem: Given a set of strings, find a shortest string that contains all of them.

   Input: Strings $s_1, s_2, ..., s_n$

   Output: A string s that contains all strings s_i as substrings, such that the length of s is minimized.

$l = 3$

ATG TGG TGC GTG GGC GCA GCG CGT



Remark. NP-hard.

Solution 1. Define overlap (s_i, s_j) as the length of the longest prefix of s_j that matches a suffix of s_i.

Construct a graph with n vertices representing the n strings $s_i, i = 1, ..., n$.

Insert edges of length overlap (s_i, s_j) between vertices s_i and s_j.

Find the shortest path which visits every vertex exactly once. This is the Traveling Salesman Problem (TSP), which is NP complete.

A slightly different problem,

Input: A set S, representing all l-mers (all l-length continuous sub-strings of a given string) from an (unknown) strings.

Output: A string s that contains all strings $s_i \in S$ as substrings.

Solution. V : $(l-1)$ length fragments (these can be obtained from our set S by considering the first and last $l-1$ characters of each fragment)

E: a directed edge (u, v) for each fragment in S that starts with u and ends with v.

An Euler path (why it is Eulerian? Note that all the l-mers are taken from a super-string) on this graph is a super-string.